Report on Successful Completion of Milestone G
Virtual California Code

Numerical Simulations for Active Tectonic Processes:
Increasing Interoperability and Performance

September 14, 2004

John B Rundle
Co-Investigator
Center for Computational Science and Engineering
University of California, Davis

Gleb Morein
Center for Computational Science and Engineering
University of California, Davis

Background: The purpose of this note is to report on the successful completion of Milestone G for the Virtual California earthquake simulation code. We were tasked with achieving a speedup of the VC simulation code of better than M/2, where M is the number of processors, on a machine having at least 256 processors. The machine was used was the Ames Lomax cluster with Turing front end.

We ran the code for 10,000 time steps and 700 fault elements (700^2 = 490,000 Green's functions). The version of code that we ran was optimized to produce multiple earthquake histories for use in computing statistics on earthquake occurrence (specifically a magnitude-frequency relation) and for eventual use in data assimilation and ensemble forecasting. Each of the code implementations on each processor generated a different earthquake history because each used different random number seeds. The number of earthquakes we compute thus scales as M, the number of processors. In evaluating performance of speedup, and since the number of events computed increases with the number M of processors, we define a "unit of computed work" as a computed earthquake having a magnitude $M \geq 5$. Thus we measured the speedup relative to the work done, so we need to define the "performance time" $\equiv P_T(M)$ of a run on M processors as the net CPU time divided by the number of $M \geq 5$ earthquakes. Thus:

$$P_T(M) \equiv (\text{CPU-net on M}) / \text{Work} \equiv (\text{CPU-net on M}) / (\#\text{ of } M \geq 5 \text{ events})$$

We then computed the speedup S(M) as:

$$S(M) \equiv P_T(1) / P_T(M)$$

This data is tabulated and plotted below.

In addition, we include instructions below for running this particular version of Virtual California.

Running Virtual California

The VC code, together with input files, can be found on:

http://hirsute.cse.ucdavis.edu/~rundle/VC_FILES/MILESTONE_G/

To run VC on Lomax cluster with Turing front end:

~~~~~~
A. Files used with VC:
~~~~~~

1. Source code:
   vcp6.f

2. Input data files:
   VC_JPL700_10.d
   VC_FAULTS_JPL.d
   VC_JPL700_out.out

3. Input parameter file:
   input10000

4. PBS template script file:
   q_vc.sh

~~~~~~
B. Before running:
~~~~~~

1. Make a test directory on Turing, for example:

   my_test and copy all files listed above to that directory.

2. Compile vc:

```
f77 -64 -O3 -o vcp6 vcp6.f –lmpi
```

3. In the my_test directory create a new directory: vc-out (for the output)

4. Create a test_run directory under Lomax temporary storage, for example:

```
/cluster/lomax/nobackup1/USER_NAME/test_run
```

and copy all files listed under A2, A3, and binary vcp6 to that directory.

5. Modify q_vc.sh to point to the temporary directory. At the bottom of the file replace:

```
cd   /cluster/lomax/nobackup1/USER_NAME/test_run
```

with the directory name created on the previous step.

6. Modify q_vc.sh for the email notification:
At the top of the file (line 9) change myname@mydomain to the email address where

you want to receive job notifications.

~~~~~~
C. Running:
~~~~~~
0. Make sure that you are in the directory you created at B1.
1. Decide on the number of CPU to be used (min 16 and max 256, on Lomax)

2. Modify q_vc.sh to replace all instances (4 instances) of the string NCPUS to the number of CPU to be used. For NCPUS = 32 you will have the following changes:

```
...
#PBS -l ncpus=32
...
/opt/mpt/mpt/usr/bin/mpirun -np 32 ./vcp6 < input10000 > oe.out

cp -f freq-mag.d $PBS_O_WORKDIR/vc-out/freq-mag32.d
cp -f oe.out $PBS_O_WORKDIR/vc-out/oe32.out
...
```

Exactly the same places should be modified for a different number of CPUS.

3. Submit your job to the Lomax queue by:

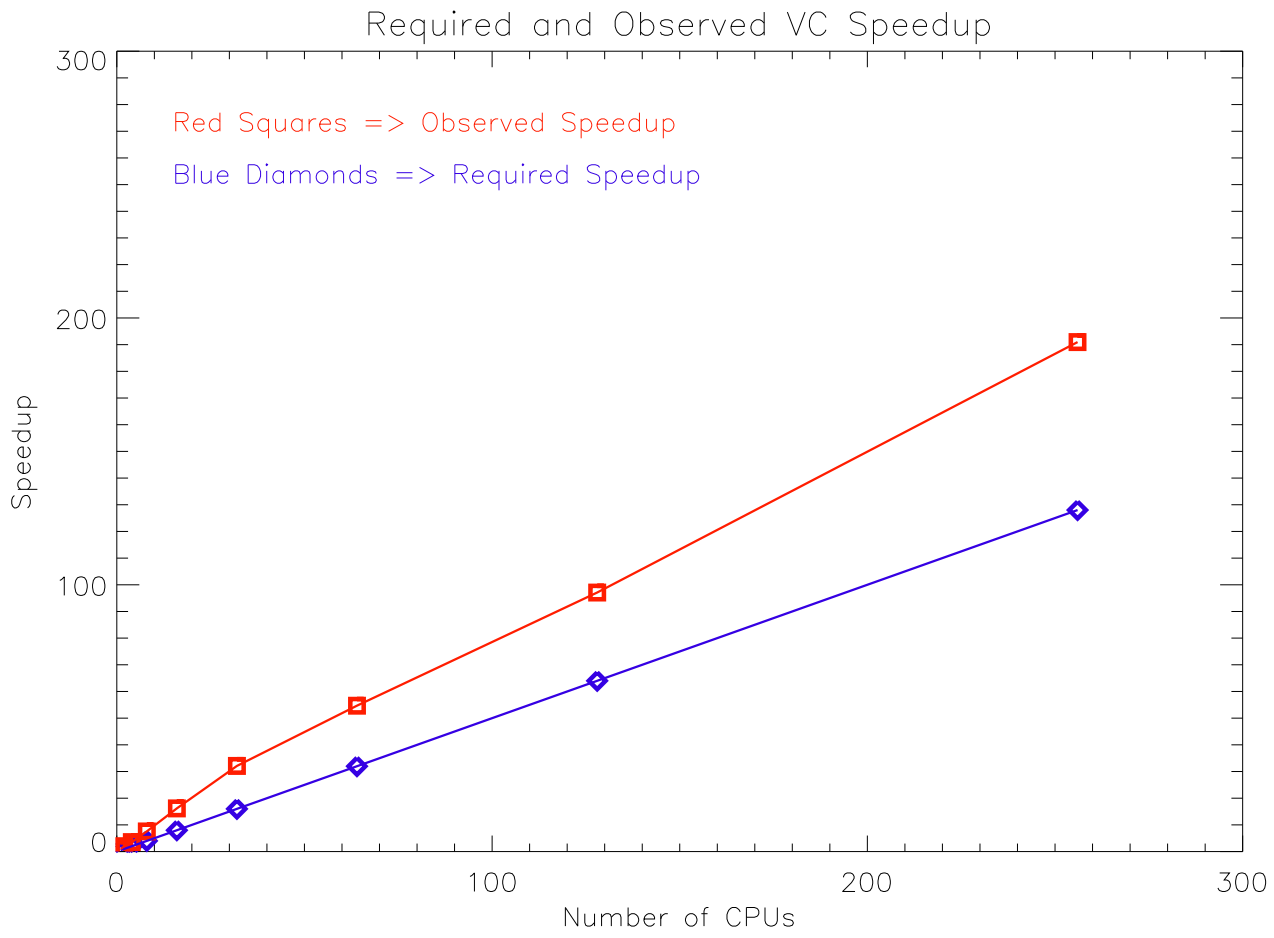
```
qsub q_vc.sh
```

You will receive an email notification of the completion (errors) of your job at the address you indicated at B6.
You can also see the status of the job using qstat command. This job runs for about 10 minutes, but the time before it starts depends on the Lomax queue load.

4. The result files (for NCPUS=32) are in vc-out directory and they are:
oe32.out - standard output and error.
freq-mag32.d - frequency-magnitude statistics for this run.

~~~~~~
D. Comments:
~~~~~~
1. Do not run concurrent jobs because files in:
/cluster/lomax/nobackup1/USER_NAME/test_run
will be overwritten.

2. File q_vc.sh contains 5 "export path" statements found to be vital and every"export" should occupy exactly one line


Required and Observed VC Speedup
Red Squares => Observed Speedup
Blue Diamonds => Required Speedup
Speedup
Number of CPUs
0
100
200
300
0
100
200
300

CPUs	Work Done	Wall Time	CPU-net Time	CPU-net/Work	Computed Speedup	Required Speedup (CPUs/2)
1	2113	456	434	0.205395173	1	0.5
2	4189	450	428	0.102172356	2.010281258	1
4	8340	506	482	0.057793765	3.553933072	2
8	16727	481	455	0.02720153	7.550868251	4
16	33462	452	425	0.012700974	16.17160769	8
32	66890	454	428	0.006398565	32.10019417	16
64	133889	536	503	0.003756843	54.67227492	32
128	267541	611	566	0.002115564	97.08768535	64
256	535550	639	576	0.00107553	190.9711541	128